

```
/**API CLIENTE/SERVIRDOR*/
```

```
void init_();  
void button_listen(int pin);  
void timer_set(int ms);  
void button_changed(int pin, int v);  
void timer_expired(void);
```

```
/**GERADOR DE EVENTOS*/
```

```
struct listElem {  
    int pin;  
    int lastVal;  
    struct listElem* next;  
};  
typedef struct listElem LIST;  
LIST* btnList = NULL;
```

```
void button_listen(int pin) {  
    LIST* novo = (LIST*) malloc(sizeof(LIST));  
    novo->pin = pin;  
    novo->lastVal = LOW;  
    novo->next = btnList;  
    btnList = novo;  
}
```

```
int now;  
int old;  
int wait;  
int exit1 = 0;
```

```
void timer_set(int ms) {  
    wait = ms;  
    //old = millis();  
}
```

```
int getWait() {  
    return wait;  
}
```

```
void loop() {  
    LIST* current = btnList;
```

```

while (current != NULL) {
    int pin = current->pin;
    int val = digitalRead(current->pin);
    if (val != current->lastVal) {
        current->lastVal = val;
        button_changed(pin, val);
    }
    current = current->next;
}

now = millis();
if (now - old >= wait) {
    old = now;
    timer_expired();
}
}

```

/**OUVINTE*/

```

int tarefa3LastBtn1 = 0;
int tarefa3ClickStep = 30;
int tarefa3State = LOW;
const int tarefa3Btn1 = 2;
const int tarefa3Btn2 = 3;
const int tarefa3Led = 4;
const int helloLed = 11;
const int helloBtn = 12;

```

```

void init_() {
    button_listen(tarefa3Btn1);
    button_listen(tarefa3Btn2);
    button_listen(helloBtn);
    timer_set(500);
}

```

```

void setup() {
    Serial.begin(9600);
    pinMode(tarefa3Btn1, INPUT);
    pinMode(tarefa3Btn2, INPUT);
    pinMode(tarefa3Led, OUTPUT);
    pinMode(helloBtn, INPUT);
}

```

```

    pinMode(helloLed, OUTPUT);
    init_();
}

void button_changed(int pin, int v) {
    int ms;
    switch (pin) {
        case tarefa3Btn1:
            tarefa3LastBtn1 = millis();
            ms = getWait();
            timer_set(ms + tarefa3ClickStep);
            break;

        case tarefa3Btn2:
            Serial.println("HEY");
            if (millis() - tarefa3LastBtn1 <= 125) {
                exit1 = 1;
                digitalWrite(tarefa3Led, HIGH);
            }
            ms = getWait();
            timer_set(ms - tarefa3ClickStep);
            break;

        case helloBtn:
            digitalWrite(helloLed, v);
            break;
    }
}

void timer_expired(void) {
    if (!exit1) {
        tarefa3State = ! tarefa3State;
        digitalWrite(tarefa3Led, tarefa3State);
    }
}

```